KNN

KNN, which stands for K-Nearest Neighbors, is a simple yet powerful algorithm used in machine learning for classification and regression. Here's a concise introduction:

Ein Bild, das Kreis, Diagramm enthält.

Automatisch generierte Beschreibung

1. \*\*Concept\*\*: KNN is a non-parametric method that relies on the idea that similar data points are likely to have similar labels. It classifies new data points based on the majority class of its 'k' nearest neighbors in the feature space.

2. \*\*How it Works\*\*:

- \*\*Distance Measurement\*\*: The algorithm calculates the distance between the new data point and all the points in the training set. Commonly used distance metrics include Euclidean, Manhattan, and Hamming distance.

- \*\*Choosing 'k'\*\*: The value of 'k' determines the number of nearest neighbors to consider. It's a hyperparameter that can be tuned.

- \*\*Majority Voting\*\*: For classification, the algorithm assigns the class that is most common among the 'k' nearest neighbors. For regression, it might take the average of the values of the 'k' nearest neighbors.

3. \*\*Advantages\*\*:

- Easy to understand and implement.

- Does not require training time as it uses the entire dataset for classification.

- Can be effective with a good choice of 'k' and distance metric.

4. \*\*Disadvantages\*\*:

- Computationally intensive as it requires calculating the distance to all points in the dataset.

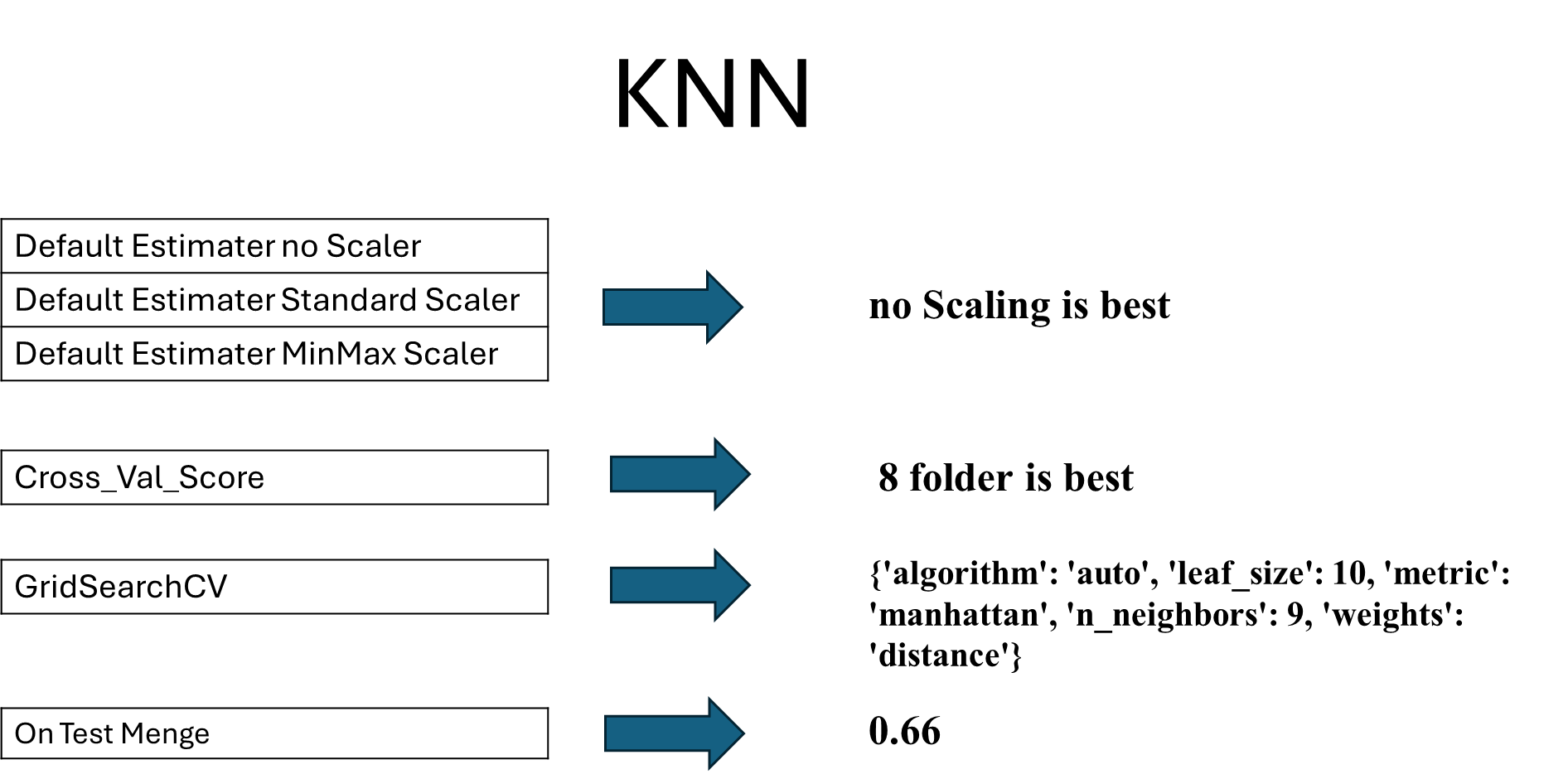
- Sensitive to irrelevant features and the scale of features.

- Requires careful selection of 'k' and distance metric.

5. \*\*Applications\*\*: KNN is used in a variety of applications, including recommendation systems, image recognition, and anomaly detection.

6. \*\*Implementation\*\*: It can be easily implemented in many programming languages and machine learning libraries, such as Python's scikit-learn library.

KNN is a versatile algorithm that, despite its simplicity, can be very effective for certain types of data and problems.



Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

SVC

Support Vector Machine (SVM) is a powerful and versatile supervised machine learning algorithm used for both classification and regression tasks. Here's a brief introduction:

Ein Bild, das Reihe, Diagramm, Schrift enthält.

Automatisch generierte Beschreibung

1. \*\*Concept\*\*: SVM is based on the idea of finding the optimal hyperplane that best separates data into different classes. The goal is to maximize the margin between the closest points of the classes, which are known as support vectors.

2. \*\*How it Works\*\*:

- \*\*Hyperplane\*\*: In a two-dimensional space, a hyperplane is a line that separates two classes. In higher dimensions, it's a hyperplane.

- \*\*Margin\*\*: The gap between the hyperplane and the nearest data points from each class is the margin. SVM aims to maximize this margin.

- \*\*Support Vectors\*\*: The data points that lie closest to the hyperplane and influence its position and orientation are called support vectors.

3. \*\*Kernel Trick\*\*: SVM can handle non-linearly separable data by using kernel functions, which allow the algorithm to operate in a higher-dimensional space without explicitly computing the coordinates in that space. Common kernels include linear, polynomial, radial basis function (RBF), and sigmoid.

4. \*\*Soft Margin\*\*: To handle noisy data, SVM introduces a soft margin that allows some misclassifications in exchange for better generalization on unseen data.

5. \*\*Advantages\*\*:

- Effective in high-dimensional spaces.

- Performs well with a clear margin of separation.

- Versatile due to the kernel trick.

6. \*\*Disadvantages\*\*:

- Can be memory-intensive with large datasets.

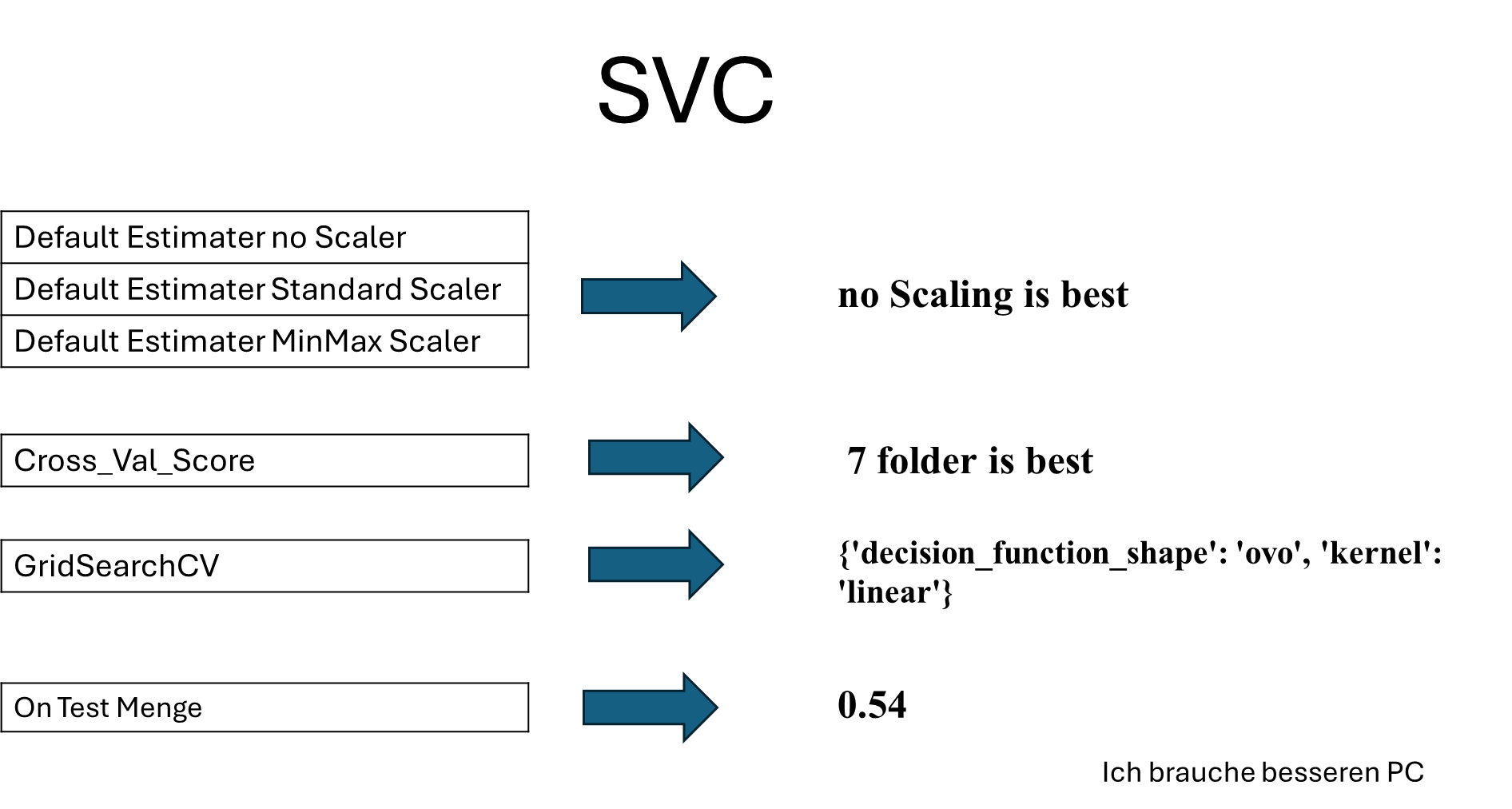
- Choice of kernel and its parameters can significantly affect performance.

- May not perform well if the data is not linearly separable without careful parameter tuning.

7. \*\*Applications\*\*: SVM is widely used in various applications, including image classification, bioinformatics, text categorization, and handwriting recognition.

8. \*\*Implementation\*\*: SVM can be implemented using various libraries and frameworks, such as scikit-learn in Python, which provides a straightforward way to apply SVM to different datasets.

In summary, SVM is a robust algorithm that can handle both linear and non-linear data separation by maximizing the margin between classes, making it a popular choice for a wide range of machine learning tasks.



Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

MLP

The Multi-Layer Perceptron (MLP) Classifier is a type of artificial neural network that is widely used for classification tasks. Here's a brief overview:

Ein Bild, das Kreis, Diagramm, Screenshot enthält.

Automatisch generierte Beschreibung

1. \*\*What is it?\*\*: An MLP is a feedforward artificial neural network that consists of multiple layers of neurons, including an input layer, one or more hidden layers, and an output layer.

2. \*\*How it works\*\*:

- \*\*Input Layer\*\*: Receives the feature vectors of the input data.

- \*\*Hidden Layers\*\*: These layers perform computations using weights and biases, and introduce non-linearity through activation functions.

- \*\*Output Layer\*\*: The final layer, which uses an activation function appropriate for the task (e.g., softmax for multi-class classification).

3. \*\*Activation Functions\*\*: Commonly used functions include ReLU for hidden layers to introduce non-linearity and softmax for the output layer to normalize output values into probabilities.

4. \*\*Training Process\*\*:

- The network is trained using labeled data by adjusting the weights and biases through backpropagation and an optimization algorithm like gradient descent.

5. \*\*Advantages\*\*:

- Capable of learning complex patterns and relationships in data.

- Flexible architecture allows for the addition of more layers and neurons to increase model complexity.

6. \*\*Disadvantages\*\*:

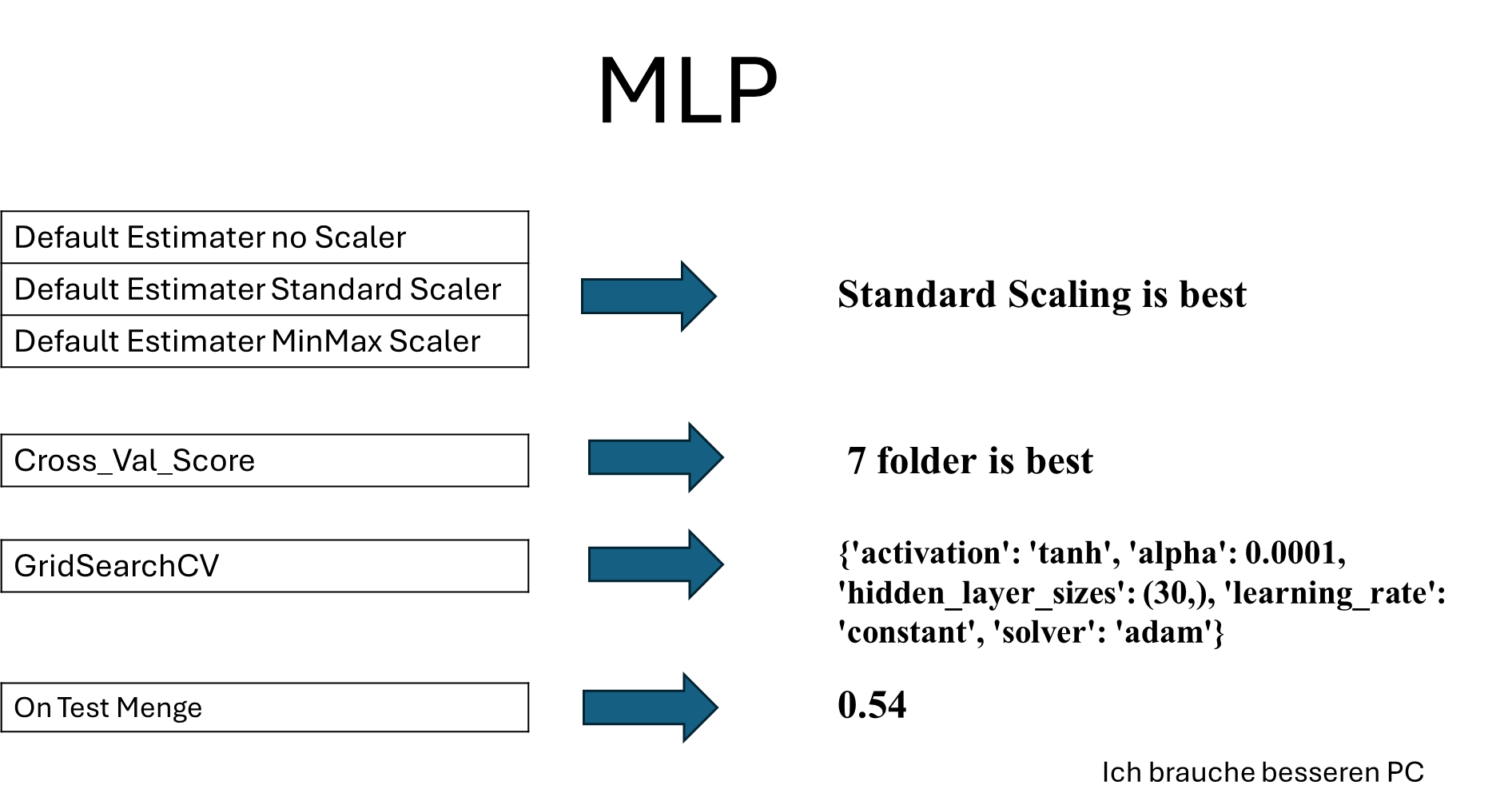
- Prone to overfitting if not regularized properly.

- Requires careful hyperparameter tuning, including the number of layers and neurons.

7. \*\*Applications\*\*: MLPs are used in various applications, including image recognition, speech recognition, and natural language processing.

8. \*\*Implementation\*\*: MLP Classifiers can be easily implemented using machine learning libraries such as TensorFlow, Keras, or PyTorch.

In summary, the MLP Classifier is a powerful tool for classification tasks, capable of learning from data and making predictions based on complex patterns.



Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

All Score

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Leasons Learned

was gut gemacht wird:

\*Teamzusammenhalt

\*Jedes Teammitglied war für seine Arbeit verantwortlich

\*Der Tutor hat auf Fragen geantwortet, wenn wir unsicher waren.

\*Wir konnten Erfahrungen für das erste Projekt mit maschinellem Lernen sammeln

was nicht gut gemacht wird:

- Ich verstehe die Parameter der einzelnen Modelle nicht genau. Ich kann die Feinabstimmung des Modells nicht sehr gut vornehmen.

- Es ist schwierig anzufangen, wenn alle zum ersten Mal ein Projekt zum maschinellen Lernen durchführt.

Was würde ich tun, wenn ich mehr Zeit und einen guten PC hätte?

\*PCA-Ergebnisse ausprobieren und umsetzen

\*mehr über die Parameter jedes Modells lernen. auf GridseachCV mehr ausprobieren.

\*Mehr Kategorie Zusammenfassung ausprobieren